ANALISA DAN PERANCANGAN APLIKASI DORMITORY MANAGEMENT MENGGUNAKAN UNIFIED SOFTWARE DEVELOPMENT PROCESS

Stanley Karouw

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Sam Ratulangi,
Jl. Kampus UNSRAT Bahu, Manado, 95115, Indonesia

E-mail: stanley.karouw@unsrat.ac.id

Abstract

Dormitory Management Web-based applications is the efficient solution for optimizing Rental Homes business process management. Customers easily get the latest information about the services provided in real-time. Owner may perform supervisory control online, which is expected to increase revenue significantly. Unified Software Development Process (USDP) methodology is a software development famework which supports use-case-driven characteristic and iterative increments, with using UML ver 2.0 as modeling tools. USDP can be used to develop software with object-oriented paradigm by actively involving all stakeholders. USDP also support business processes optimization and shorten application development

Keywords: *Applications Software, Object Oriented, USDP, UML, Software Development*

Abstrak

Aplikasi Dormitory Management berbasis Web adalah solusi efisien untuk mengoptimalkan proses bisnis pengelolaan Rumah Sewa. Pelanggan dapat dengan mudah mendapatkan informasi terkini tentang layanan yang diberikan secara real-time. Owner dapat melakukan kontrol pengawasan secara online. Sehingga diharapkan dapat meningkatkan pendapatan secara signifikan. Metode Unified Software Development Process (USDP) merupakan metode pengembangan perangkat lunak yang use-case driven dan increment-iterative, dengan perangkat UML ver 2.0 untuk keperluan pemodelan. USDP dapat digunakan untuk mengembangkan perangkat lunak yang dibangun dengan paradigma berorientasi obyek dengan melibatkan semua stakeholders secara aktif. Karakteristik metode USDP yang use-case driven, increment-iterative ini dapat memfokuskan pengembangan perangkat lunak pada fungsionalitas aplikasi, sehingga dapat mendukung optimalisasi proses bisnis dan mempersingkat waktu pengembangan aplikasi.

Kata Kunci: *Aplikasi, Berorientasi Obyek, USDP, UML, Pengembangan Piranti Lunak*

1. Pendahuluan

Konsep usaha rumah sewa merupakan konsep hunian kos yang modern dan dilengkapi dengan berbagai fasilitas pendukung guna memberikan kenyamanan bagi para penghuni dengan fasilitas seperti kamar mandi, AC (Air Conditioner), koneksi internet, *laundry and cleaning service*. Perkembangan jenis usaha ini semakin menjamur di wilayah perkotaan yang dekat pusat-pusat pendidikan tinggi maupun menengah. Fenomena ini semakin terlihat menjelang masa tahun ajaran baru. Untuk meningkatkan *market share,* layanan dan menjadikan usaha rumah sewa identik dengan rumah sewa modern, maka pemanfaatan Teknologi Informasi (TI) dalam mendukung proses bisnis unit usaha ini menjadi suatu pilihan.

Pengelola unit usaha Rumah Sewa menjalankan proses bisnisnya secara manual sehingga sangat tidak efisien dan optimal. Ditemukan banyak permasalahan yang terkait dengan keakuratan data untuk *report, booking* (pemesanan) dan pendataan aset. Berbagai masalah ini mempengaruhi neraca *cost-benefits* yang ada. Selain itu sistem yang ada sekarang ini sangat rentan terhadap faktor kesalahan manusia *(human error*) baik disengaja maupun tidak

disengaja. Hambatan juga ditemukan guna meningkatkan tingkat hunian karena ketatnya persaingan. Hal ini berakibat pada turunnya pendapatan. Biaya yang dikeluarkan untuk pemasangan iklan (melalui media cetak) menjadi semakin besar. Tambahan lagi, *owner* Rumah Sewa, sering bepergian keluar kota, sehingga sulit untuk melakukan kontrol pengawasan terutama mengenai *report income* setiap bulan berjalan.

Aplikasi Dormitory Management berbasis Web bertujuan untuk:

1) Meningkatkan kontrol pengawasan pengelolaan rumah kos oleh owner
2) Memudahkan pembuatan laporan oleh manager.
3) Meningkatkan *market share.*
4) Memberikan kemudahan akses informasi rumah kos.

Implementasi aplikasi ini pada akhirnya diharapkan dapat meningkat tingkat hunian, sehingga pada akhirnya meningkatkan pendapatan secara kontinyu. Disamping itu, kenyamanan penghuni juga tetap diperhitungkan.

2. Unified Software Development Process

Pada dasarnya pengembangan atau rekayasa perangkat lunak dapat berarti menyusun aplikasi yang benar-benar baru atau – yang lebih sering terjadi – menyempurnakan yang telah ada sebelumnya. Sehingga, dapat dikatakan bahwa mengembangkan perangkat lunak merupakan suatu proses berkelanjutan. Inilah prinsip pengembangan perangkat lunak. Untuk mengembangkan perangkat lunak secara berkelanjutan, diperlukan suatu kerangka kerja. Kerangka kerja pengembangan perangkat lunak ini memandang perangkat lunak sebagai suatu produk yang dihasilkan melalui suatu proses logis yang berurutan dengan masukan-masukan yang tepat. Inilah yang dimaksudkan dengan pendekatan *process framework*.[1] System Development Life Cycle (disingkat SDLC) atau disebut Daur Hidup Pengembangan Sistem adalah kerangka kerja proses pengembangan perangkat lunak yang berkelanjutan. SDLC membagi tahapan pengembangan perangkat lunak menjadi[1]: Tahap Komunikasi, Tahap Perencanaan, Tahap Pemodelan, Tahap Konstruksi dan Tahap Implementasi

USDP merupakan salah satu kerangka kerja pengembangan perangkat lunak berorientasi obyek. USDP, yang kemudian lebih dikenal dengan *Unified Process* (disingkat UP), dikembangkan oleh Graddy Booch, Ivar Jacobson dan James Rumbaugh[2], ditujukan untuk secara konsisten beradaptasi dengan tren pengembangan perangkat lunak yang semakin besar dan semakin kompleksnya. USDP merupakan kerangka kerja pengembangan yang berbasiskan komponen; ini berarti bahwa perangkat lunak yang dihasilkan akan terdiri atas komponen-komponen perangkat lunak yang saling terhubung melalui antarmuka yang terdefinisi dengan baik.[2][3]

Karakteristik kerangka kerja USDP dalam mengembangkan perangkat lunak disebut *use-case driven proces*. Ini mengandung pengertian bahwa USDP terus berpedoman pada diagram Use-Case, yang mengilustrsikan fungsionalitas sistem dalam gambar yang mudah dipahami oleh user. Diagram Use Case merupakan diagram sentral yang menjadi pedomen model pengembangan USDP lainnya. Model pengembangan USDP lainnya adalah[2][3]:

1) Model Analisis; yang bertujuan untuk memperhalus dan merinci definisi-definisi masing-masing use-case;
2) Model Perancangan, yang digunakan untuk mendefinisikan struktur statis aplikasi seperti kelas, antarmuka dan hubungannya masing-masing dalam kerangka perangkat lunak yang dibangun;
3) Model Konstruksi, yang memuat komponen-komponen (merepresentasikan kode-kode dalam bahasa pemrograman tertentu yang dipilih) dan melakukan pemetaan kelas-kelas ke komponen-komponen;
4) Model Implementasi, yang berguna untuk mendefinisikan simpul-simpul komputer secara fisik dan melakukan pemetaan masing-masing komponen ke setiap simpul komputer yang ada;
5) Model Pengujian, yang ditujukan untuk mendeskripsikan skenario testing untuk melakukan verifikasi dan validasi terhadap perangkat lunak yang dikembangkan.

USDP menggunakan UML (Unified Modelling Language) sebagai alat pemodelan perangkat lunak yang dikembangkan. UML merupakan suatu metode *modelling* generasi ketiga dan bahasa spesifikasi yang sifatnya *non-proprietary*. UML adalah suatu metode terbuka yang digunakan untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifak-artifak dari suatu pengembangan sistem *software* yang berbasis pada obyek[4]. UML merupakan hasil kompilasi *best engineering practice* yang sudah terbukti sukses dalam pemodelan sistem yang besar, sistem yang kompleks, khususnya pada level arsitektural.

Ada tiga aspek utama dalam pemodelan sistem yang mampu didukung oleh UML[2]:

1) *Functional Model*, untuk menunjukkan fungsionalitas dari suatu sistem dari sudut

pandang *user* atau pengguna. Ini dicapai dengan menggunakan *Use Case Diagram.*

2) *Object Model,* untuk menunjukkan struktur dan substruktur dari suatu sistem dengan menggunakan object, atribut, operasi dan juga asosiasi. Ini dicapai dengan menggunakan *Class Diagram.*
3) *Dynamic Model,* menunjukkan internal behavior dan suatu sistem. Ini dicapai dengan menggunakan *Sequence Diagram, Activity Diagram* dan juga *Statechart Diagram.*

3. Metodologi Pemecahan Masalah

Fokus penulisan paper ada pada proses analisis dan perancangan aplikasi. Tahapan analisis dan perancangan yang dilakukan adalah sebagai berikut:

1) Menentukan persyaratan; dengan menggunakan metode wawancara. Kemudian dengan menggunakan User Stories, Persyaratan dikelompokkan menjadi persyaratan fungsional dan non fungsional.
2) Pemodelan Fungsional; dengan menggunakan Diagram Use Case.
3) Pemodelan Obyek/Struktur; dengan menggunakan Diagram Kelas.
4) Pemodelan Behaviour; dengan menggunakan Diagram Sequence.
5) Perancangan; dilakukan pada lapisan managemen data, managemen desain antar-muka dan desain arsitektur fisik.

Untuk kepentingan penulisan paper ini, maka proses perencanan proyek dan pemodelan bisnis tidak dilakukan. Penulisan juga dibatasi hingga tahap membuat model rancangan aplikasi, sehingga tidak mencantumkan *coding* program

4. Pembahasan

Proses analisis dan perancangan aplikasi dengan menggunakan USDP dimulai dengan mendefinisikan/menetukan persyaratan (atau *requirements*). Persyaratan perangkat lunak dibedakan menjadi persyaratan fungsional dan persyaratan non-fungsional[5]. Fokus pengembangan aplikasi ada pada persyaratan fungsional. Persyaratan fungsional ini yang akan dimodelkan melalui Diagram Use Case. Metode yang digunakan dalam mengumpulkan persyaratan fungsional adalah dengan melakukan wawancara terhadap user dan pengguna aplikasi kemudian membuat User Stories. Setelah itu, pengembang bersama user mengklasifikan persyaratan aplikasi berdasarkan user stories tersebut.

Beberapa ketentuan fungsional yang harus dipenuhi oleh sistem antara lain sebagai berikut:

1) Booking
   a) 1.1 Sistem dapat menampilkan informasi mengenai tempat kos (foto kamar, lokasi, fasilitas, dan daftar harga sewa kamar).
   b) 1.2 Sistem dapat menangani pemesanan kamar secara online (*booking online*).
   c) 1.3 Sistem dapat membatalkan pemesanan kamar yang telah di *booking* oleh calon penghuni.

2) Payment
   a) 2.1 Sistem dapat menangani pembayaran uang sebagai tanda jadi memesan kamar (*down payment* sebesar 10%) secara *online* via bank.
   b) 2.1 Sistem dapat menangani pembayaran uang sewa dan mengkonfirmasinya ke penghuni rumah kos.

3) Room Management
   a) 3.1 Sistem harus dapat menyimpan dan menampilkan data penghuni, kamar yang dihuni serta layanan yang dipesan.

4) Report
   a) 4.1 Sistem dapat menyimpan dan men-*generate* laporan keuangan bulanan dan tahunan.
   b) 4.2 Sistem dapat mencatat pengeluaran harian.
   c) 4.3 Sistem dapat men-*generate* aset report.

Untuk persyaratan non-fungsional dibedakan dalam persyaratan operasional, performansi dan kemanan. Beberapa ketentuan non-fungsional yang harus dipenuhi oleh sistem antara lain sebagai berikut:

Ketentuan Operasional (lingkungan fisik dan teknis sistem yang diaplikasikan):

1) Sistem dapat dioperasikan pada *PC Dekstop* dan *Notebook* serta terlihat dalam resolusi display 1024x768 dan 800x600.
2) Sistem harus dapat bekerja pada semua *web* browser seperti *internet explorer* dan *mozila firefox.*
3) Sistem harus dapat diakses pada sistem operasi Windows dan Linux.
4) Sistem harus dapat diakses pada komputer dengan spesifikasi *hardware* minimal, yakni Hard Disk 1 GB, Memori RAM 128 MB, dan Prosesor Pentium standar.

Ketentuan Performansi (kecepatan, kapasistas, dan keandalan):

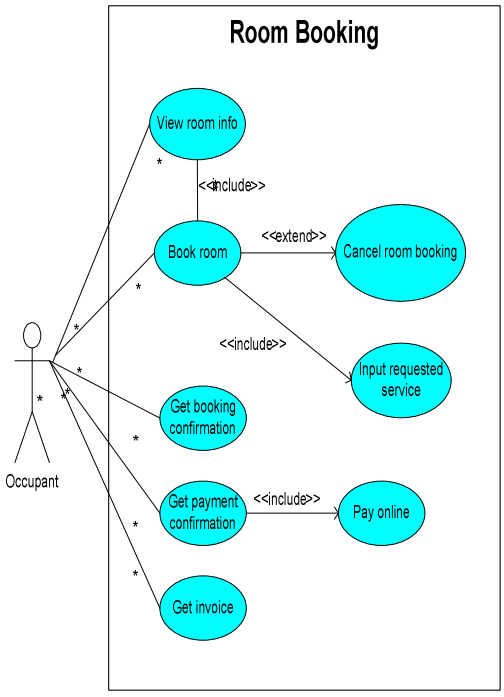
1) Setiap interaksi sistem dengan user tidak boleh lebih lama dari 3 detik.
2) Sistem harus dapat digunakan atau dioperasikan dalam 24 jam dalam sehari, 7 hari dalam seminggu dan 356 hari dalam setahun.
3) Sistem dapat menangani x transaksi secara bersamaan.

4) Sistem harus dapat men-*generate historica*l data laporan keuangan selama x bulan.

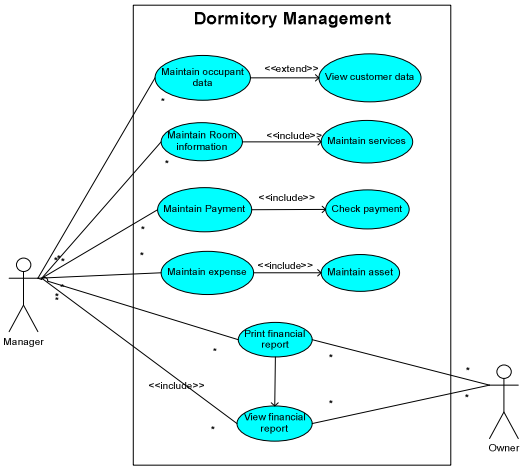
Ketentuan Keamanan (akses otorisasi);

1) Sistem harus memiliki sistem otorisasi bertingkat, dalam hal ini dibedakan menjadi otorisasi *owner, manager,* dan otorisasi penghuni atau calon penghuni.
2) Sistem harus mengatur otorisasi untuk penghuni dan calon penghuni ; tidak dapat mengakses dan mengubah serta meng-*update report income* harian, bulanan, dan tahunan.
3) Sistem harus mengatur otorisasi untuk *manager*; dapat meng-update laporan bulan berjalan dan atau tahun berjalan, namun tidak dapat mengubah laporan yang telah lewat
4) Sistem harus mengatur otorisasi untuk *owner*; dapat memeriksa dan mengubah laporan bulanan atau tahunan yang berjalan maupun yang telah lewat.

Pemodelan fungsional digambarkan dengan Use Case Diagram. USDP menekankan pada model use case. Use Case akan menggambarkan semua fungsionalitas yang akan dibangun dalam aplikasi. Use Case Diagram akan menjadi pedoman untuk langkah-langkah analisis dan disain selanjutnya. Berikut ini adalah Use Case Diagram dari aplikasi Dormitory Management berbasis Web (lihat Gbr. 1 dan Gbr.2)

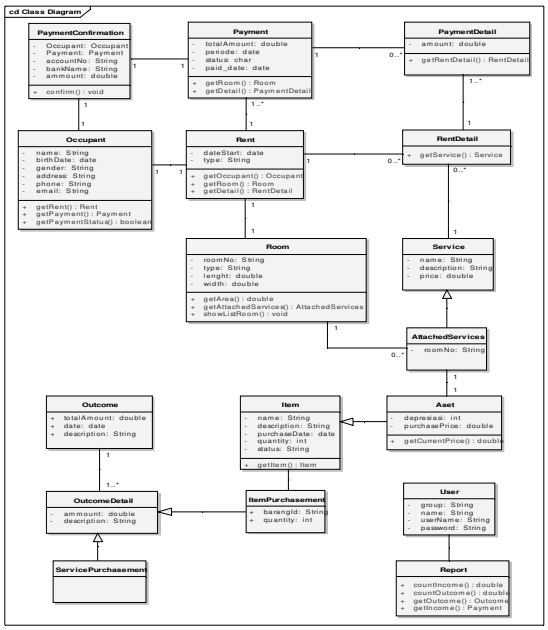


Gbr. 1. Use Case Room Booking



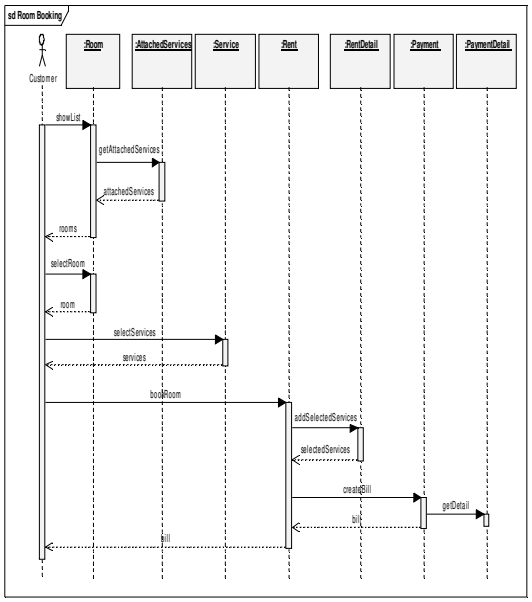
Gbr. 2. UML Use Case Dormitory Management

Diagram Kelas (lihat Gbr. 3) merupakan model statik yang menggambarkan semua kelas dan hubungan di antara kelas yang bersifat konstan dalam sistem sepanjang waktu. *Building block* utama dari kelas diagram adalah kelas, yang menyimpan dan mengelola informasi dalam sistem. Selama fase analisis, kelas mengacu pada *people, place, events,* dan suatu sistem yang menangkap informasi. Pada fase desain dan implementasi, kelas mengacu pada *artifacts* seperti *windows, form,* dan objek lain yang digunakan untuk membangun sistem. Atribut dari sebuah kelas dan nilainya menggambarkan keadaan sebuah objek yang dihasilkan dari sebuah kelas, sedangkan perilakunya diwakili oleh *operation*[5].

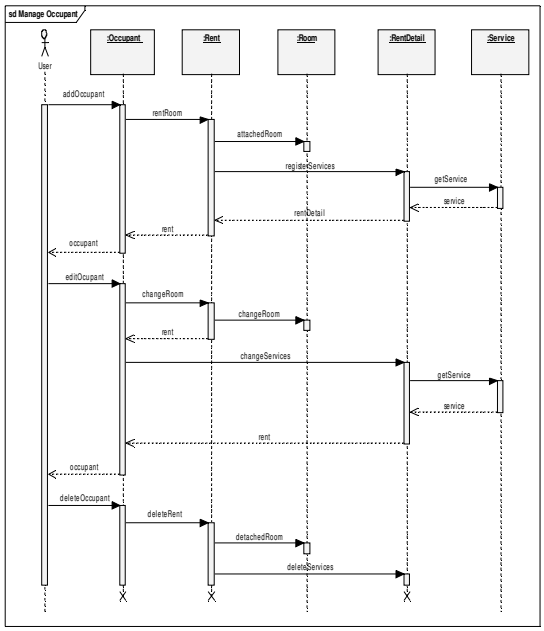


Gbr. 3. UML Class Diagram

Pemodelan behaviour disajikan dengan menggunakan Diagram Sequence (lihat Gbr. 4 dan Gbr. 5). Berikut adalah diagram sequence dari beberapa proses utama dalam aplikasi.

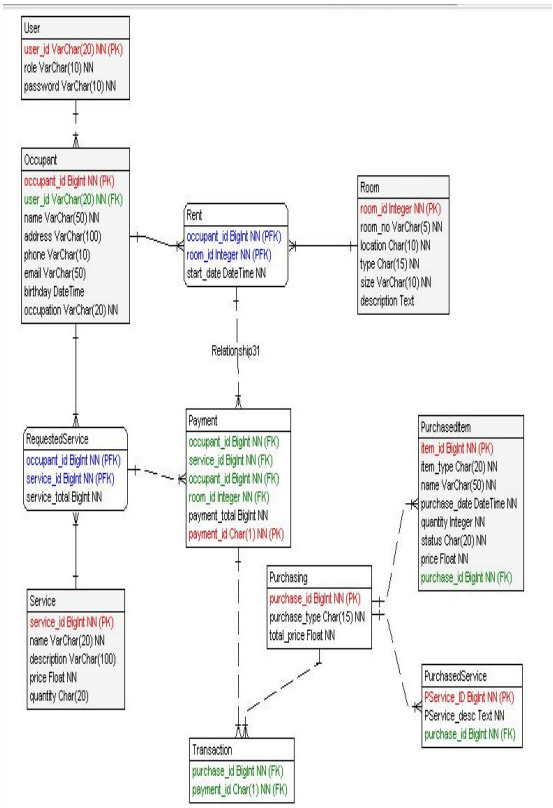


Gbr. 4 UML Sequence Diagram Skenario Room Booking



Gbr. 5 UML Sequence Diagram Skenario Manage Occupant

Desain *data management layer* termasuk akses data *manipulation logic* beserta *desain actual* dari *storage*. Komponen *data storage* dari *data management layer* mengatur bagaimana data harus disimpan dan digunakan oleh program yang menjalankan sistem. Terdapat 4 (empat) tipe dasar dari *object-persitence format* yang bisa digunakan untuk menyimpan obyek sebuah sistem aplikasi, yaitu *sequential (random), object-oriented database,* dan *relational database*.[5] Pada penelitian ini format yang dipakai adalah RDBMS *(relational database). Mapping Problem Domain Object to RDBMS* digambarkan dalam ERD berikut. (Lihat Gbr. 6)



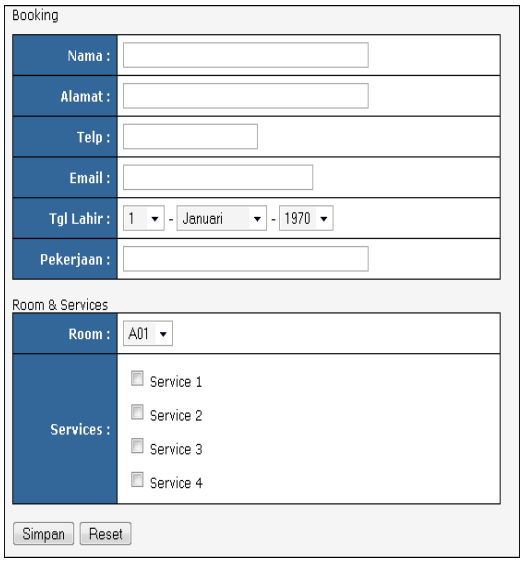
Gbr. 6 Mapping Problem Domain Object to RDBMS

Untuk Tabel *Relationship* memuat data relasi antar tabel yang terdapat pada system. Tabel tersebut dapat dilihat pada Gbr. 7 dibawah ini.

Relationship type	Parent entity	Child entity	Card.
Non-identifying	RequestedService	Payment	1:N
Non-identifying	Rent	Payment	1:N
Identifying	Occupant	Rent	1:N
Non-identifying	Payment	Transaction	1:N
Identifying	Room	Rent	1:N
Non-identifying	Purchasing	PurchasedService	1:N
Identifying	Occupant	RequestedService	1:N
Non-identifying	User	Occupant	1:N
Identifying	Service	RequestedService	1:N
Non-identifying	Purchasing	PurchasedItem	1:N
Non-identifying	Purchasing	Transaction	1:N

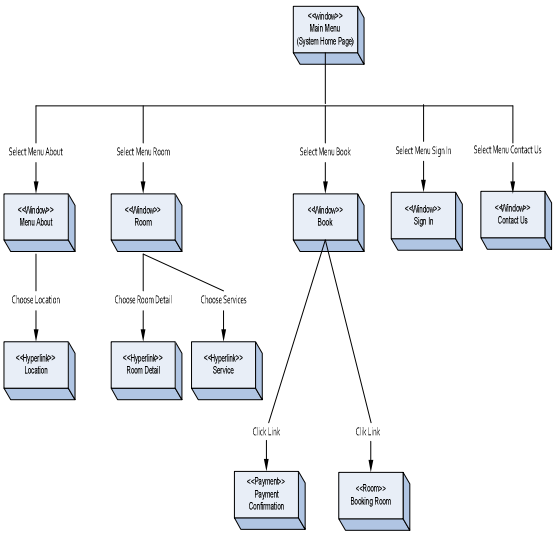
Gbr. 7 Relasi Antar Tabel

Setelah membuat model-model perangkat lunak, selanjutnya adalah mendesain lapisan antar muka pengguna (atau *interface design*). *Interface design* merupakan proses pendefinisian bagaiman sistem berinteraksi dengan unit eksternal, seperti *customer, suplier,* dan sistem yang lain. Antarmuka Pengguna terdiri dari 3 (tiga) bagian dasar. Yang pertama adalah *navigation mechanism,* berupa cara *user* memberi instruksi ke sistem dan memberitahu sistem apa yang harus dilakukan, (misalnya *buttons, menus*). Yang kedua adalah *input mechanism,* berupa cara sistem menangkap informasi (misalnya *form* untuk menambahkan informasi calon penghuni rumah kos yang baru. Yang ketiga adalah *output mechanism* berupa cara sistem menyediakan informasi bagi user atau ke sistem yang lain (misalnya laporan, *web pages*). Untuk keperluan penulisan paper ini maka hanya akan ditampilkan salah satu i*nterface design prototyping* aplikasi Dormitory Management berbasis Web yang terdiri dari beberapa tampilan bagi *user,* salah satunya adalah *form booking* yang diperlihatkan pada Gbr. 8 di bawah ini:



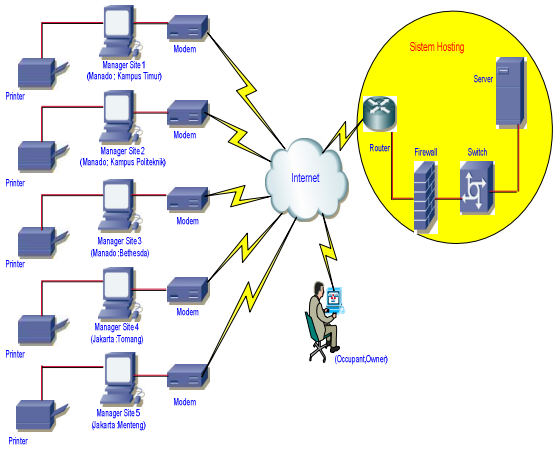
Gbr. 8 *Form* yang digunakan untuk proses *booking* kamar dan pengisian *service* kamar yang diinginkan oleh calon penghuni

Untuk gambar *design navigasi* dari *web*site yang akan dilihat *visitor,* dapat dilihat pada Gbr. 9 berikut.



Gbr. 9 *Navigation Design*

Lapisan Desain *Physical Architecture* aplikasi Dormitory Management berbasis Web ini menggunakan metode *Client-Server Achitecture* untuk menjaga keseimbangan proses antara *client* dan *server* yang memiliki fungsi aplikasi masing-masing. *Client* bertanggung jawab atas *presentation logic* sedangkan *server* bertanggung jawab atas *data access logic* dan data *storage.* Desain *Physical Architecture* tersebut dapat dilihat pada Gbr. 11 dibawah ini:



Gbr.10 *Physical Architecture Layer Design*

Pada Gbr. 11 diatas, dapat dilihat bahwa strategi perancangan infrastruktur aplikasi Dormitory Management berbasis Web ini menggunakan sistem *hosting*, melalui paket *hosting* yang ditawarkan oleh salah satu penyedia jasa *hosting* yang ada di Indonesia.

Alasan menggunakan sistem *hosting* adalah:

1) Tidak perlu membeli infrastruktur jaringan seperti *router, firewall, switch* dan *server.*

2) Dengan tidak adanya *network infrastructure* tersebut, maka management pengelolaan jaringan (seperti *maintaning network infastructure*) tidak perlu dilakukan. Disamping itu skema koneksi internet 24/7 tidak perlu ada.

3) Menekan *operational cost*.

5. Kesimpulan

Beberapa kesimpulan dari penulisan paper ini antara lain adalah:

1) Metode Unified Software Development Process, yang disingkat USDP atau UP, dapat digunakan untuk membangun Aplikasi berbasis Web dengan pendekatan berorientasi obyek.

2) Metode USDP merupakan metode pengembangan perangkat lunak yang *use-case driven, incremental* dan *iterative*. Karakteristik metode ini mengharuskan keterlibatan *stakeholder* secara aktif.

3) Kakas UML versi 2.0 dapat memodelkan perangkat lunak secara terperinci dan sederhana, sehingga sangat bermanfaat untuk pengembang dan pengguna. Namun demikian, dibutuhkan ketelitian pengembang dalam memilih diagram-diagram UML yang tepat bagi pengembang dan mudah dipahami oleh user; untuk efektivitas setiap proses yang terkait analisis dan disain perangkat lunak yang akan dikembangkan.

4) Proses penentuan persyaratan merupakan proses yang sangat penting untuk dilakukan. Proses ini harus melibatkan semua *stakeholder* yang terkait dalam pengembangan aplikasi. Keterlibatan user bersifat mandatory saat proses ini dilakukan.

5) Aplikasi Dormitory Management berbasis Web dapat meningkatkan pengawasan owner, memudahkan site manager membuat laporan sehingga pada akhirnya mengoptimalkan efisiensi proses bisnis pengelolaan Rumah Sewa.

Referensi

[1] Roger Pressman, *Software Engineering, A Practitioner's Approach*, 6th Ed, McGrawHill, Singapur, 2005

[2] Stephen Schach, *Object Oriented Software Engineering*, 8th Ed, McGrawHill, 2008

[3] Adi Nugroho, *Rekayasa Perangkat Lunak berorientasi Objek dengan Metode USDP*, Penerbit Andi, Yogjakarta, 2010

[4] Martin Fowler, *UML Distilled; Panduan Singkat Bahasa Pemodelan Obyek Standar*, Edisi ke-3, Penerbit Andi, 2005.

[5] Dennis, A., Wixom, B.H., and Tegarden, D.,*System Analysis and Design with UML Version 2.0: An Object-Oriented Approach*, 2nd ed. Wiley International Edition, 2005.